

California State University, San Bernardino

CSUSB ScholarWorks

Theses Digitization Project

John M. Pfau Library

2004

Intranet concept for small business

Allen Gregg Lenaburg

Follow this and additional works at: <https://scholarworks.lib.csusb.edu/etd-project>



Part of the [Business and Corporate Communications Commons](#)

Recommended Citation

Lenaburg, Allen Gregg, "Intranet concept for small business" (2004). *Theses Digitization Project*. 2710.
<https://scholarworks.lib.csusb.edu/etd-project/2710>

This Project is brought to you for free and open access by the John M. Pfau Library at CSUSB ScholarWorks. It has been accepted for inclusion in Theses Digitization Project by an authorized administrator of CSUSB ScholarWorks. For more information, please contact scholarworks@csusb.edu.

INTRANET CONCEPT FOR SMALL BUSINESS

A Project
Presented to the
Faculty of
California State University,
San Bernardino

In Partial Fulfillment
of the Requirements for the Degree
Master of Business Administration

by
Allen Gregg Lenaburg

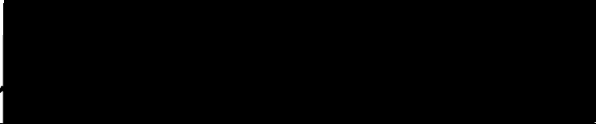
June 2004

INTRANET CONCEPT FOR SMALL BUSINESS


A Project
Presented to the
Faculty of
California State University,
San Bernardino

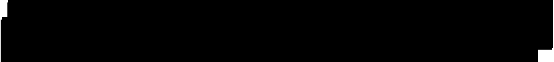
by
Allen Gregg Lenaburg
June 2004

Approved by:


Dr. Tony Coulson, Chair
Information And Decision Sciences

5/25/04
Date


Dr. Jake Zhu


Dr. Walter Stewart

ABSTRACT

An intranet is a tool that can facilitate the exchange of information throughout an organization. Historically, intranets have been the domain of large corporations because they had the financial resources to accommodate the high costs of designing, implementing and maintaining the intranet applications. Today, because of new technology, software and services, the cost of implementing an intranet is now within the grasp of many small businesses. This project contains a working intranet application dedicated to small business use and is designed to present a compromise between scratch building and purchasing an intranet application. A business that implements an intranet similar to the one described in this project will have the ability to publish and retrieve documents in an intuitive manner. Additionally, by using the event calendar functionality employees, from anywhere they have an Internet connection, will be able to access company date-driven events in real time. Furthermore, this project will describe how to place an intranet onto a leased web hosting service or on a previously existing network.

ACKNOWLEDGMENTS

I wish to thank the following for their invaluable contributions to this project:

- My wife, Lisa, for her non-compromising support.
Without her this project would never have materialized.
- Dr. Sunny Baker, Dr. Jake Zhu, and Dr. Tony Coulson who pointed me in the right direction and patiently reviewed my material to keep me on course.
- The teaching staff at California State University San Bernardino for creating an atmosphere of learning and for sharing their knowledge and experiences.
- To the employees of Mercy Air Inc. for allowing me to test my applications on them, and for providing me with invaluable feedback.

TABLE OF CONTENTS

ABSTRACT	iii
ACKNOWLEDGMENTS	iv
CHAPTER ONE: INTRODUCTION	
Statement of the Problem	1
Purpose	1
CHAPTER TWO: LITERATURE REVIEW	
Introduction	3
Intranet Goals	4
Intranet Cost Benefits	5
Successful Applications	6
Small Business Intranets	9
Solutions	10
Conclusion ...	12
CHAPTER THREE: PROJECT DESCRIPTION	
Introduction	13
Web Server Options	14
Performance	15
Security	16
Site Navigation	17
Administration	18
Database	18

Project Applications	19
Summary	23
CHAPTER FOUR: IMPLEMENTING AN INTRANET	
Introduction	24
Planning for Success	24
Implementation	25
Promoting an Intranet Site	26
User Acceptance	27
Summary	28
CHAPTER FIVE: CODING METHODOLOGY	
Introduction	29
Active Server Pages	29
Server-Sided Scripting	30
Client-Sided Scripting	31
Summary	31
CHAPTER SIX: PROJECT SUMMARY	32
APPENDIX A: DIAGRAMS OF WEB SERVER OPTIONS AND INTRANET NAVIGATION MENUS	34
APPENDIX B: PROJECT SCREEN SHOTS	37
APPENDIX C: PROJECT SOURCE CODE	42
APPENDIX D: DATABASE STRUCTURE	83
REFERENCES	89

CHAPTER ONE

INTRODUCTION

Statement Of The Problem

An intranet utilizes the same technology that runs the Internet. It is implemented within the confines of a business structure, and is specifically designed to meet the information needs of that business (Durval, 2002). Historically, larger businesses and corporations were the only organizations that had the resources to implement an intranet. However, with the recent developments of newer technology, services, and lower costs, the potential for intranet use by smaller businesses has increased. Small businesses may benefit from an intranet because of its ability to effectively streamline the processes for retrieving and distributing information.

Purpose

The purpose of this project is to build a working intranet containing core applications that create the framework for a small business intranet. Moreover, this intranet can be customized specifically to the needs of a particular business. This project is designed to be a compromise between building an intranet application from

scratch and purchasing a fully operating software package. This intranet model is intended for students, faculty, business owners and managers who are interested in a working intranet that is performance based and scalable.

CHAPTER TWO

LITERATURE REVIEW

Introduction

The research approach used in this project focuses on reviewing the current literature to explain important concepts contributing to the success of intranets in use by larger companies. The objective is to adapt these concepts so that smaller businesses can make them suitable to their scale of operations. The scope of the research will include defining the goals of an intranet, the cost effectiveness of implementing an intranet, and defining important core intranet application components. Specifically, based on case research, applications including information retrieval, employee services, and document publishing will be examined. Moreover, the research will explore the potential of intranet usage for small businesses including the potential market, obstacles to intranet acceptance, and solutions to overcoming those obstacles. Additionally, the solutions will describe the lower cost thin servers and the hosted web service options. Much of the research consists of peer-reviewed articles and case studies that analyze businesses utilizing an intranet

in some form, and companies that are marketing intranet solutions to small businesses.

Intranet Goals

The goal of an intranet is to manage the information that a company has obtained over time and making it accessible to the organization as a whole (Starnes, 1997 p. 31). This management of information facilitates information sharing allowing the decision makers and problem solvers within the company to do their jobs more effectively. With the sharing of information comes collaboration. "When collaboration occurs, two or more individuals with complementary skills interact to create a new, shared understanding" (Guengerich, Graham, Miller, McDonald 1997, p.20). Guengerich et al., goes on to describe (p.21) that a knowledge management system should give everyone access to company information, and it should be available 24 hours a day across time and space, and does not require users to be computer experts. An intranet conforms well to the definitions of a knowledge management system because it never sleeps, is accessible globally via the Internet, and it employs a simple point and click interface.

Intranet Cost Benefits

Research on intranet cost effectiveness shows that in 1996 over 60% of the Fortune 500 companies were utilizing some form of an intranet (Fulk, Hinds, Hollingshead, Kiesler, Monge, 2002). Larger corporations have found that one of the reasons that intranets are successfully implemented is because that they use the same proven technology as the Internet (Pitt, Murgolog-Poore, Dix, 2001, p. 106). Adopting this technology is cost effective because the expenses of research and development have already been incurred, and most of the technical problems have previously been solved. The Internet protocol of TCP/IP and HTML are universal and have been providing a means for global information sharing that corporations are exploiting on a large scale. Therefore, the costs of developing an intranet are less, compared to other applications, because the technologies for an intranet are derived from the Internet. Software, expertise and the interface of HTML already exist, and are widely available (Tittle, 1997, p.100-101). McGaughey (1999, p.7) describes how Internet technology is not constrained by organizational and environmental boundaries. Moreover, it

can present various forms of information in a universal format that enhances the value of the information. Many companies use intranets as a means to improve their business model by managing their information resources effectively. By doing this the processes that are information critical become more streamlined and accessible. McGaughey (p.7), describes that when information is posted on an intranet in a timely manner it is readily available to any computer in the world that has corporate intranet access.

Implementation of an intranet has resulted in a significant ROI for some larger corporations. For example the Mitre Corporation reported a \$62 million decrease in operating costs by investing \$7.2 million in a corporate intranet (Ward, 2001, p.3). Additionally, according to Cisco Systems, they extensively post and use all business related information on their intranet, and because of their diligence they saved \$800 million in costs (Ward, p.1)

Successful Applications

Companies that have intranets utilize them in a variety of ways, however, there are some intranet

applications that the literature has shown to be consistently successful.

Information Retrieval

One concept of an intranet that has been successful involves using methods that streamline the process of searching for information, and consequently leaving more productive time for other tasks. For example, Delta Airlines implemented an intranet consisting of online maintenance manuals and parts catalogs that resulted in a 50% reduction in time that maintenance crews spent searching for information. This replaced the tedious process of manually searching through volumes of manuals. Therefore, the critical time for aircraft servicing turnaround was reduced allowing planes to fly sooner and generate more revenue (Deimler, Hansen, 2001, p.97). The Charles Schwab firm has reduced costs and increased productivity by making information easily available with an online system that includes a powerful search engine. Consequently the core processes are more streamlined and employees have increased efficiency in performing tasks (Deimler, Hansen, p.98).

Employee Services

Another intranet application utilizes an intranet for empowering employees to be more self-sufficient with the ability to handle internal administrative tasks without over-burdening staff resources. Delta Airlines improved its crew scheduling system to provide better matching of crew and flight demands while reducing costs. Therefore, allowing the employees the convenience of accessing crew schedules over the Internet (Deimler, Hansen, p.97). Cisco uses their intranet for employee self-service matters such as ordering equipment, or modifying benefit options. As a result employee related tasks that previously required processing by administrative staff over several days is greatly reduced (Deimler, Hansen 2001, p.98).

Publishing

Many successful intranet strategies pertain to the publication of information. Companies are now converting paper documents to digital format and publishing them on the intranet for employee or customer use. According to Devendra (1997, p.8), costs are reduced because less staff is needed to process these documents compared to the more labor intensive methods of printing and mailing documents, forms, and handbooks. The process of updating a document

on an intranet occurs in real-time and therefore employees can be confident they will all have the same current information.

Small Business Intranets

Obstacles

The intranet has been in existence for several years, but it is mostly utilized by larger companies that could afford the purchase, maintenance, and operational requirements for operation. Smaller companies have been reluctant to embrace the intranet technology essentially due to the associated costs of purchasing equipment, hardware, and the burdens of system maintenance. The more powerful servers and network systems are appropriate for larger corporations who have the technical resources to maintain them. Often, they are not suitable for small businesses that cannot afford the adequate resources (Jones, 2000, p.67).

Potential Market

The advancements in recent technology has significantly reduced the costs of servers required to run an intranet site. Today the typical desktop computer has the computing power that only powerful and expensive

servers were capable of just a few years ago. Additionally, the emergence of Internet service providers that are dedicated to web-hosting services have also lowered costs significantly. These web-hosting services are able to spread the costs of running a robust web server among several customers and therefore making web hosting affordable. Historically, small businesses would not purchase intranet packages because of costs associated with hosting a web site (Sanders 1999 p. 81). However, with the lowered costs of newer technology and services, smaller business can now afford their own intranet solutions. As a result the market for small business intranets has opened.

Solutions

Evaluating Technology

With the availability of thin servers and web hosting services, the infrastructure required to run an intranet has become far less expensive and with much less complexity. According to Roberts (2002), small businesses are placing budget priority on technology especially in the area of Internet applications. Furthermore, they are now re-evaluating the potential for technology and how it can be suitably integrated into the organization.

Lower Cost Servers

Thin servers are stand-alone computers that use mostly desktop components and are less expensive than their network server counterparts. Jones (p.67) describes that small and mid-size businesses cannot afford the high costs associated with maintaining general-purpose servers and will find the thin servers a cost effective alternative. Furthermore, these server appliances can be connected to the Internet with DSL or other broadband technologies to provide multiple employee connections. A smaller-scale web server can then be added to function as the core of a company intranet.

Web Hosting

A web hosting service is an Internet service provider that has hosting web sites as its primary focus. According to Sanders (p.81), these sites provide access to powerful database servers such as Microsoft SQL, scripting languages, and tools that make administration of the web site much easier. The costs can be less than \$100 a month. The web hosting service can allow a company to establish a business intranet without purchasing any network or server equipment. The only equipment requirements are an Internet connection, web-page editor, and browser.

Conclusion

The literature has shown that many large organizations have found an intranet to be an effective means to share and manage information. An intranet is a cost effective solution for information management because it is based on the same technology that powers the Internet, and utilizes universal protocols to send and receive data. Moreover, corporations have implemented intranet systems in a variety of ways including improving information retrieval, empowering employees, and publishing documents. Furthermore, with the development of lower cost hosting technology and software, a potential market has opened, and smaller businesses now have the opportunity to implement intranet solutions.

CHAPTER THREE

PROJECT DESCRIPTION

Introduction

The research in this project indicates that many large corporations utilize intranets in a variety of ways to manage their information effectively. However, many small businesses have found the requirements for expensive equipment and dedicated staff resources to be cost prohibitive for implementing their own intranet sites. The intranet contained within this project is designed to overcome those obstacles by designing all applications to function on a single web server that also contains a database. This allows an intranet to be compact and either installed on a web hosting service or attached to a previously existing network. All of the applications are web-based and the requirements for performance, security, site navigation, and administration are all included in the package. This intranet contains an event calendar, methods for publishing, and document searching and retrieval as its core applications.

Web Server Options

This project is designed to work on a Microsoft Internet Information Server with the .Net Framework version 1.1 or greater installed. To create greater flexibility there are two choices pertaining to how the intranet is installed on a web server. The intranet can be placed on a leased web hosting service or on a local network attached to a broadband Internet connection. (See Appendix A, Figure 1).

Web Hosting

The leased web hosting service requires a monthly fee to utilize the space on their computers usually with a variety of options. The least expensive option is offered as a shared web server where several customers are provided space on the same web server. The dedicated server is another web hosting option where the customer is leasing the entire web server. The dedicated web server usually costs more and the customer usually must provide their own software applications. The better hosting services will have database applications installed as a standard option. The primary advantages for the small business that elects to lease from a web hosting service are minimal capital costs due to purchasing equipment and software, and the

staffing to maintain the web site. In contrast, the web server performance is dependent upon the infrastructure of the web hosting service and the amount Internet traffic that it can bear at any given time. Consequently, a small business may have significant performance problems if it finds that the web hosting service has inadequate bandwidth relative to the amount of customers sharing a web sever.

Network Server

The second option for installing an intranet is to use a thin web server that is attached to an existing network. The thin web server has the advantage of being accessible from either the network or through the Internet (See Appendix A, Figure 1). A company with more sensitive information will find this a more secure choice since network authentication will be available. In comparison to the web hosting service the network server option will require additional technical resources to implement and maintain the web server.

Performance

Performance is a primary objective of this project and a great deal of server-sided scripting is utilized to make the site dynamic. The Microsoft frameworks of Active

Server Pages (ASP and ASP.Net) are used because of the large availability of tools and designing resources available. To improve performance the user interface is uncluttered with an intuitive functional layout, and consists of very few non-essential graphics since they affect the web page loading time at the browser. Most users of an information-based intranet desire the ability to quickly obtain the desired information and are not interested in waiting for a dazzling visual display to load. In his article about intranet winning characteristics, Ward (2001), states that the web site should be designed with consideration to the user who has the slowest Internet connection. The goal of this project is for web pages to load in less than 6 seconds on a standard dial-up Internet access account.

Security

This intranet project has a built-in username and password system of security since most leased hosting services do not provide for system authentication. The database contains a table that is dedicated to user accounts. Users are assigned the privileges of a standard-user, publisher, or administrator. The standard user can

view documents and fill out online forms only. The publisher has the ability to upload documents to the web server. The administrator can view and modify user accounts. Entry to the intranet is through a username and password screen. If desired, the administrator can provide users with a code that will allow them the ability create their own username and password. The username and password are sent to the web server in text format. Consequently, a company should not place sensitive information on this intranet when using the leased hosting service. Companies with more sensitive information should place the intranet on a local network that provides for more secure forms of authentication.

Site Navigation

One of the objectives of this project is to create an uncluttered intuitive interface for the users. Therefore, a menu bar is located at the top of each page to facilitate site navigation See Appendix A, Figure 2). Menus and sub-menus are displayed when the mouse is moved over the menu bar. This style of navigation is advantageous over users trying to find numerous hyperlinks placed on the page. However, this type of menu system requires the

client-side scripting language of Javascript and some browsers may have this feature turned off or they are not compatible. Consequently there is a site map page that does display hyperlinks for site navigation.

Administration

Managing an intranet is important, and this project provides the ability to administer some aspects of the intranet using a web browser on a remote client computer. To accomplish this ease of administration, server-sided scripting is used to create a site maintenance section accessible through an Internet Browser. User accounts can be accessed and modified by using the authentication menu selection. The intranet usage for each user is also available for display. The section for viewing the log allows the intranet manager to retrieve a list of users that accessed the intranet during a particular day. The database backup selection essentially downloads the database to the client computer.

Database

A database is required for this project and Microsoft Access is the database currently used, but connecting any other ODBC compliant database is an option.

One advantage of Microsoft Access is that the database file can be easily downloaded to a client computer for use as a source of data backup, and for simplifying changes to the database design. As an option, for businesses that expect a high usage, the intranet project has been tested with a Microsoft SQL Server successfully. The database contains tables that are used to store data for the event calendar, user accounts, documents and training rosters. Relationships are established between tables, where required, to maintain referential integrity and utilize cascading updates and deletes. Web pages connect to the database by using server sided scripting and SQL query statements.

Project Applications

The applications designed into this prototype intranet are intended to make the site dynamic so that employees will have access to the most current organizational information. Additionally, the publishers for this intranet can upload documents and modify event calendars without having to adjust any hyperlinks. These applications include an event calendar, and sections for document upload, storage, and retrieval.

Event Calendar

The event calendar has the primary function of retrieving events from a database, and displaying them in a monthly calendar format. Operations and a training calendar are utilized, but other calendars can be added based on the needs of the company. Events are displayed on the calendar as hyperlinks, and when clicked by the mouse will display details pertaining to that event. Moreover, if the event falls under the category designated as training, a roster is created so that users have the opportunity to register for that particular class roster online. The instructor can place a limit on the size of the roster, and can view the class roster at anytime. Publishers have the ability to add and modify events for the calendar by using the calendar menu option or by clicking an existing calendar event and then selecting the edit button. Users can search the event database by selecting predefined criteria. In this project, events can be searched by category, and the month they fall under. Additionally, the calendar can be further searched by keywords in the event title or a description that matches the pattern entered by the user. The search results will display a list of hyperlinks that point to the event

details page where an expanded view of a calendar event is shown. Also, when publishing an event under the announcements category, the event is not placed on the event calendar, but is displayed separately on the announcements page so that non-date related information can also be published.

Publishing

Another application that is designed into this intranet project is dedicated to the process of publishing documents or files to the server from a client computer using only a web browser. The publishing feature is significant to this project because it provides a simple system where employees can share information, and collaborate. This method of publishing is more streamlined compared to using the established methods of mailing, faxing, and printing. The publishing section provides an innovative layer of security between the user and the server. Users do not have actual access to the server files, but must utilize the database as the mechanism to perform limited file operations. Consequently, only file types that meet pre-determined criteria are permitted on the server. This file restriction is essential to block malicious and dangerous files from uploading to the server.

The database is synchronized with the files located on the server, and contains descriptive information that can be used when searching for particular files. Furthermore, the user files can be kept in a single restricted folder because the database maintains the file organization structure. The publishing page displays a browse button that permits the user to select a file from their computer for uploading to the server. The input fields of category, author, and description are required and will be appended to the database during the upload process.

Document Search And Viewing

The document-viewing section provides a means to streamline the efforts employees' use to search for and download documents and files that have been placed on the server. The viewing section uses the database to carry out search and download functions. In the view documents section, input fields selected by the user define the search parameters. This feature can be especially useful for companies that generate numerous memos. Files can be queried and sorted according to category, author, date, and description. After clicking the search button, a list of matching files is displayed to the user. The author and description criteria allow for the use of keyword patterns.

For example, entering the author name of Jan will return files that have author names of Jane, Janet, and Janie.

Summary

The intranet contained within this project is designed to significantly reduce the investment in capital and staff resources normally required to implement and maintain the corporate style intranet found in larger companies. For the smaller business, this intranet can be installed on a web hosting service or an existing network. It contains all the applications in a web-based format that are required to provide good performance, easy administration, and an intuitive user interface.

CHAPTER FOUR

IMPLEMENTING AN INTRANET

Introduction

The implementation of a successful intranet requires proper planning to ensure that the project has support from both the users and management. Prior to placing the intranet into production it must be fully tested to catch errors that might cause loss of data or user dissatisfaction. Additionally, the intranet must be promoted to the users as a tool that has usefulness and is easy to use because their acceptance and usage will ultimately decide the fate of the project.

Planning for Success

A successful intranet is a readily available and trusted tool for information exchange so that users can make effective business decisions (White, 2001, p.50). An intranet will have a better chance of long-term success if it is designed and promoted as a strategic asset to the company, and is supported by management. Therefore, an intranet should have a manager who is considered the sponsor for the project, and will provide both political and financial support (Robertson, 2002, p.1).

Furthermore, steps should be taken to ensure that the intranet is reliable, and accurate. Policies should be written that describe the process and intervals of data backup, the procedures for disaster recovery, and how content will be managed (Robertson, p.3).

Implementation

The intranet application documented in this project should run once the web server has been configured correctly, but some adjustments will be required before deploying the site into full production. Therefore, it is highly recommended that this intranet be placed on a development or staging computer first. This enables the developer to check for errors and to make the site customized to the organizational needs. Guengerich, et al. (p.314) suggests three sequential methods of testing. Starting with unit testing where only one function or transaction is committed while observing for errors. System testing involves testing the whole system prior to production, and integrated testing checks performance on the production platform.

A good practice is to let users of various computer skill levels access the intranet, and provide constructive

feedback before deployment. Additionally, each hyperlink and menu options should be followed to verify that there are no dead-ends. All fields on forms should be tested using various values to determine if any patterns of input are rejected by the database. Multiple back-up copies of the intranet files and database should exist before any modifications are attempted.

Promoting an Intranet Site

The process of promoting an intranet site begins early in its development. One concept is to consider the users as co-owners, and they "buy in", thereby describing their needs and concerns pertaining to the intranet content. Another concept is to treat the users as customers, and respond to their needs in a timely manner. Ward (2001) describes the importance of promoting the intranet to the user to be a fundamental winning intranet characteristic, and that it should focus on the needs of user, and how their processes can be integrated into the application. The goal is to achieve organizational-wide acceptance of the intranet by the users (Smith, 2001, p. 3). It is also important to monitor the content that is placed on the intranet including information that is

outdated. Material that is outdated will make searches less effective, and may cause decision makers to use the wrong data. (Smith, 2001, p.3).

User Acceptance

The gains in performance by implementing any innovative software can be offset by the lack of user acceptance. Two important variables for user acceptance are the perceived usefulness of an application and its perceived ease of use (Davis, 1989, p. 319). Perceived usefulness pertains to the user's perception of how well an application is enhancing their performance or is beneficial to the organization. Perceived ease of use is related to the amount of effort that is required to perform functions on an application. An easy to use application allows users to allocate the limited resource of effort to other tasks. According to Davis (p. 329) the relationship between perceived usefulness and ease of use are significantly correlated to the usage of an application. Therefore, it is important in the design and implementation processes to take into account the usefulness of an intranet. Moreover, how the intranet relates to business processes and enhances users performance when completing assigned tasks.

Additionally, intranet design and implementation should consider the degree of difficulty experienced and the amount of effort expended when a user is interfacing with an intranet.

Summary

An intranet will have greater chance of success if it is a strategic asset to the organization. Therefore, both management and users must properly support it. Moreover, an intranet needs to be reliable so that data integrity remains intact and users will not abandon the project because of frustration. Furthermore, the intranet administrators must promote the site by treating the users as valued customers who expect timely responses to their needs and problems. Finally the intranet should be perceived as useful and easy to use by the users to gain their acceptance of it.

CHAPTER FIVE

CODING METHODOLOGY

Introduction

An intranet is comprised of many web pages that are rendered to the client from the server upon request. Additionally, an intranet utilizes scripting code embedded in the web pages to carry out instructions that make the intranet site dynamic. The embedded scripting codes can perform many processes in the background such as responding to user inputs, manipulating the screen display, or submitting queries to a database. Some of the scripts are performed on the server using Microsoft's Active Server Pages (ASP), while other functions occur on the client computer using the client-sided scripting language of Javascript.

Active Server Pages

Static web pages found on the majority of web sites are useful for displaying information that does not change often. However, the goal of an intranet is to provide the users with current information to facilitate processes and decision-making. This project utilizes ASP to a great extent because an ASP contains scripting code that is

processed on the server before sending information to the client. It is this server-sided processing that is the mechanism for connecting the web page output to a database, and making an intranet dynamic. The dynamic intranet can display current data in real time, in contrast to the static web site where web pages are updated manually with any new information. Furthermore, ASP is more secure because it sends only the hypertext markup language (HTML) to the Internet browser, therefore, the source code remains on the server and is not revealed to the user.

Server-Sided Scripting

The server-sided scripting language used predominately in this project is Microsoft Vbscript and it is a subset of the Microsoft Visual Basic (VB) programming language. Fortunately, the Internet has a an enormous amount of Vbscript samples that demonstrate most of the functions that are required to create dynamic web pages. Some of the web pages on this intranet project also utilize the new .Net Framework released by Microsoft. The .Net pages have the full Visual Basic language available, including features that reduce the amount of programming code required to perform a task. Furthermore, the code can be

pre-compiled into assembly code for better performance, and security.

Client-Sided Scripting

Sometimes it is advantageous to have functions performed on the client computer without having to send information to and from the server. Javascript is the client-side scripting language utilized on this project. Javascript is a scripting language that is widely used on the Internet, and is compatible with most Internet browsers. Javascript can act upon events generated by the user's computer and can be very useful for validating user input on the browser before sending any data to the server. Client-sided scripting can also display dialog boxes and windows that can contain various messages to the user.

Summary

For the intranet to function dynamically the web pages must be able to display the information to the client based upon user inputs, and changes to a database. The process of using scripting code in the web pages can be accomplished on the server or performed on the client computer.

CHAPTER SIX

PROJECT SUMMARY

The literature has shown that large companies have utilized an intranet to facilitate the sharing of information, and it has proved to be a cost effective means to streamline many processes. These larger companies have found that an intranet is effective as a tool to reduce the time expended searching for information. Furthermore, an intranet can be an instrument for empowering employees to complete routine administrative tasks while minimizing the use of staff resources. Finally, an intranet can be used to publish documents more efficiently than printing, faxing, and mailing.

The small business market is now ready to benefit from intranet technology because of the declining costs and improved capabilities of newer services. A smaller business can now implement an intranet with minimal costs and manpower requirements by using either the thin web server or a hosted web service provider.

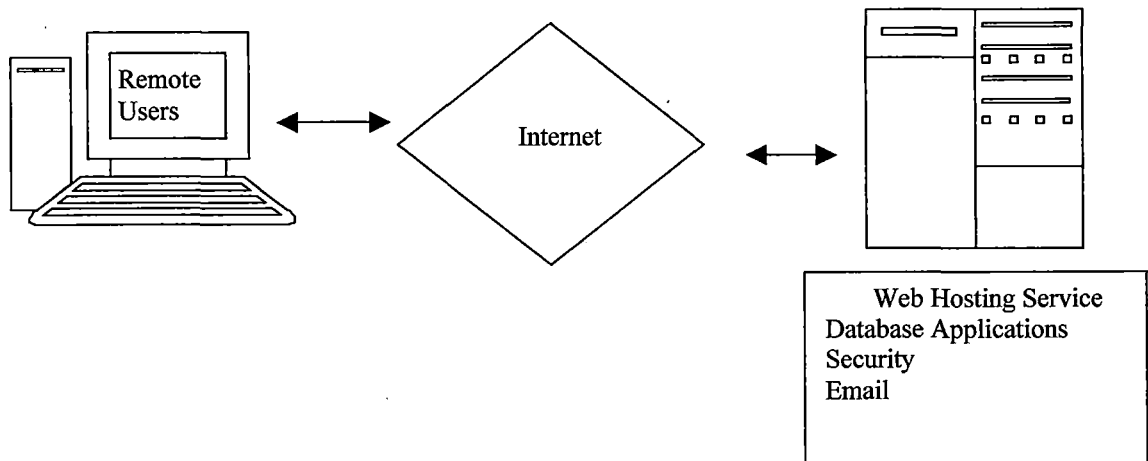
The intranet included as part of this project is designed to introduce a framework upon which a small business can build a customized intranet without purchasing

a software application or designing from scratch. This intranet is suited to the company that desires a central location for document publishing and an intuitive event calendar that can be available 24 hours a day globally. This project is database driven using server-sided scripting, that creates an intranet that is dynamic so it can deliver current information to the employees.

APPENDIX A

DIAGRAMS OF WEB SERVER OPTIONS AND
PROJECT NAVIGATION MENUS

Web Hosting Service



Local Area Network

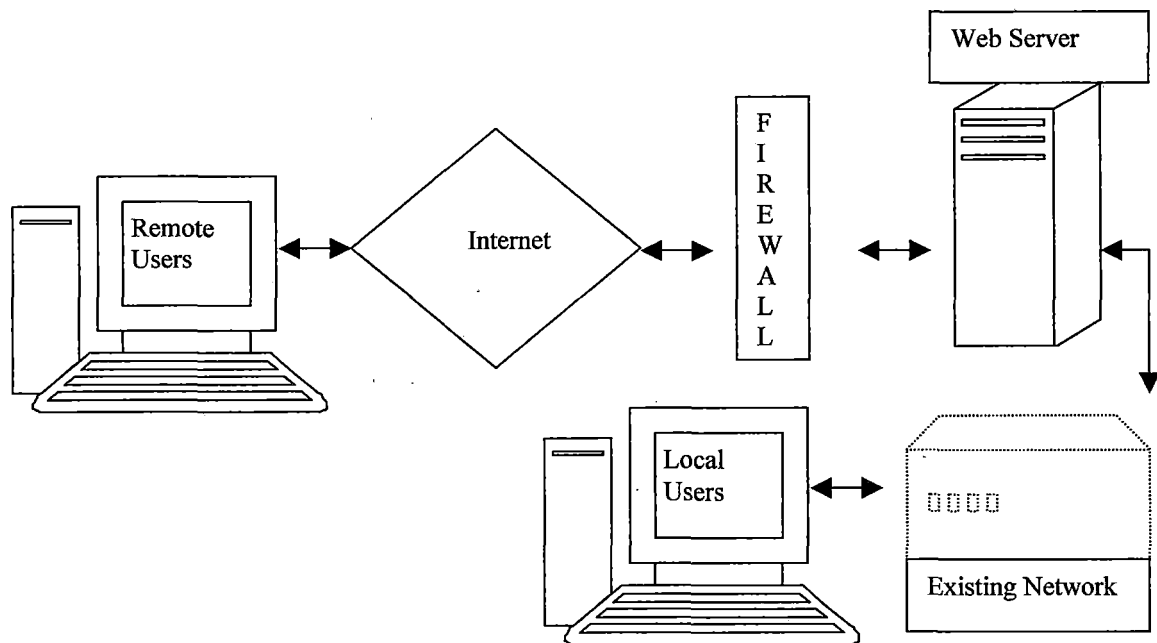


Figure 1 Web server Options

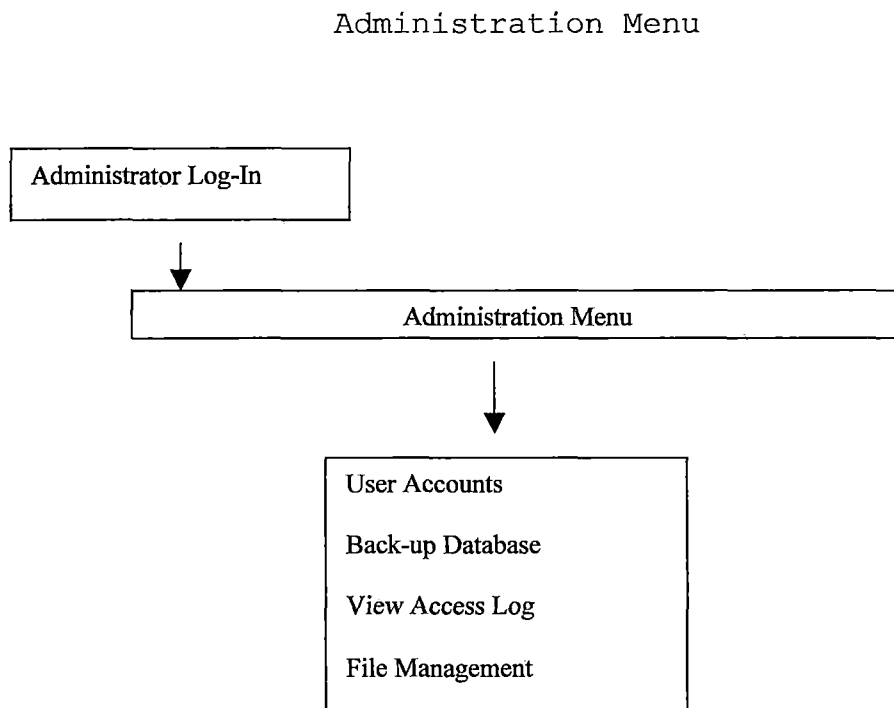
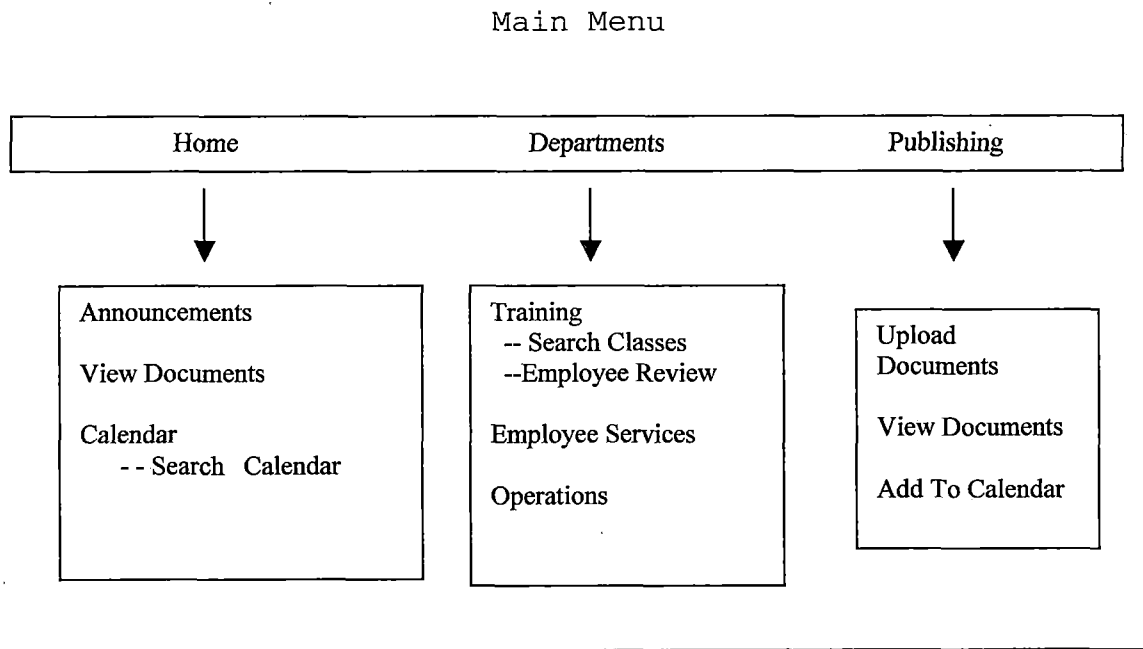



Figure 2 Navigation Menus

APPENDIX B
PROJECT SCREEN SHOTS

[Home](#)
[Departments](#)
[Publish](#)

[Announcements](#)
[View Documents](#)
[Calendar](#)



L

L

L

Ranch

[Site Map](#)
[Email Webmaster](#)

Intranet Web Site

TODAY'S EVENTS

Calendar


Today *September 21, 2003*

TIME	TITLE	Category
1:00 PM	Childrens Clinic	Training
3:00 PM	Dressage Clinic	Training

Figure 3 Main Page

Home	Departments	Publish
------	-------------	---------

[Site Map](#) [Email Webmaster](#)



L
L Ranch
L

Intranet Web Site

Document Publishing

Select File to Upload:


Category: Description:

Author ☐ Lockout

Status:

Figure 4 Publishing Page

Home	Departments	Publish	
------	-------------	---------	--


L
L
L **Ranch**

[Site Map](#) [Email Webmaster](#)

Intranet Web Site


View Documents

Search Criteria

Category: All
 Author Keyword:

Month: All
 Year: 2003
 Description Keyword:


Sort By:
☒ Filename
 ☐ Category
 ☐ Date
 ☐ Author


All

	Filename	Author	Description	
Download	handbook.htm	Jane Doe	Employee Handbook 2003	Delete
Download	midterm.xls	me	xl program	Delete
Download	Delete

Figure 5 View Documents Page

INTRANET WEB SITE



L
L Ranch
L

PLEASE ENTER USERNAME

PASSWORD

[Email Webmaster](#)

Figure 6 Authentication Page

APPENDIX C
PROJECT SOURCE CODE

Editing Calendar Events

File Name: Event_Main.aspx

Description: Displays a form for creating or editing existing calendar events, and submits to a database.

Language: Vb.net

Code

```
<%@ Page Language="VB" Debug="true" Inherits="lifenetconn.lifenetdbconn" %>
<%@ import Namespace="System.Data" %>
<%@ import Namespace="System.Data.SqlClient" %>
<%@ import Namespace="System.IO" %>
<%@ outputcache location="none" %>
<%@ import Namespace="System.web.mail" %>
<script runat="server">

'Event Maint
'By Allen Lenaburg

Sub Page_Load(sender As Object, e As EventArgs)
    *****
    Dim ex as exception
    if not ispostback then

        'Restrict non publishers
        if session("auth")="view" then
            button1.visible=false
            fdate.readonly=true
            title.readonly=true
            'linkdoc.enabled=false
            category.enabled=false
            author.readonly=true
            description.readonly=true
            subbutton.visible=false
            verase.visible=false
        end if 'session auth=view

        Dim res as sqlconnection
        Dim pwds as string="n411al" 'needed for dll file
        Dim sqlCommand As New SqlCommand()
        Dim datareader As sqldatareader
        Dim querystring as string

        res=getquery(pwds)
        res.open()
        sqlCommand.connection=res

        'populate the dropdown with file names from flood
        querystring="SELECT filename from flood"
        sqlCommand.commandtext=querystring

        try
            datareader=sqlcommand.executereader()
        catch ex
```

```

response.write("Error Retrieving From Database")
senderror(ex)
res.close
response.end
end try

linkdoc.datasource=datareader
linkdoc.datatextfield="filename"
linkdoc.datavaluefield="filename"
linkdoc.databind()
datareader.close
dim myli as new listitem
    myli.text="None"
    myli.value="None"
    linkdoc.items.insert(0,myli)

'if existing record
if request("event_id")<>" " then
    event_id.text=request("event_id")
    old.checked=true
    qrystring="select * from calender_event where event_id=" & event_id.text

    sqlcommand.commandtext=qrystring

try
    datareader=sqlcommand.executereader()
catch ex
response.write("Error with database")
senderror(ex)
res.close
finally
end try

    if datareader.read()=true then
        title.text=datareader.item("name")
        fdate.text=datareader.item("start_dt")
        author.text=datareader.item("author")
        description.text=datareader.item("description")
        category.Items.FindByValue(datareader.item("category")).Selected = true
        linkdoc.Items.FindByValue(datareader.item("linkfile")).Selected = true
    end if 'datareader is true
    datareader.close

end if 'id <> " "

'clean up
res.close
sqlcommand.dispose()
sqlcommand=nothing
res.dispose()
res=nothing

if request("evdte")<>" " then fdate.text=request("evdte")
subbutton.attributes("onclick")="DoCal(this.form.fdate);"

end if 'not postback

*****
' if the category is a meeting then display register link
if category.selecteditem.text="Meetings" and old.checked=true then
    register.NavigateUrl="roster.aspx?event_id=" & event_id.text
    register.visible=true
end if 'category=meetings
ireturn.NavigateUrl="page1.aspx?date=" & fdate.text

End Sub 'page load

```

```

*****
Submit Button
*****

```

```

Sub Button1_Click(sender As Object, e As EventArgs)
    Dim ex as exception
    Dim res as sqlconnection
    Dim pwds as string="n411al" 'needed for dll file
    Dim sqlCommand As New SqlCommand()
    Dim refnumber as integer
    Dim rand as new random()
    Dim datareader As sqldatareader
    dim querystring as string
    dim added as integer=0

    if old.checked=true then
        querystring="UPDATE [calender_event] set name=@name,[start_dt]=@startdate" & _
            ",author=@author,category=@category,description=@description"& _
            ",linkfile=@linkfile"& _
            " WHERE event_id=" & event_id.text
    else
        refnumber=rand.next(1000,100000)
        ref.text=refnumber
        querystring="INSERT INTO [calender_event](name,start_dt,author,category,description,linkfile,refnumber)"& _
            " VALUES(@name,@startdate,@author,@category,@description,@linkfile,@refnumber)"
        added=1
    end if 'checkbox is true

    sqlCommand.Parameters.Add("@name", System.Data.SqlDbType.Char).Value = title.text
    sqlCommand.Parameters.Add("@startdate", System.Data.SqlDbType.Char).Value = fdate.text
    sqlCommand.Parameters.Add("@author", System.Data.SqlDbType.Char).Value = author.text
    sqlCommand.Parameters.Add("@category", System.Data.SqlDbType.Char).Value = category.selecteditem.text
    sqlCommand.Parameters.Add("@description", System.Data.SqlDbType.Char).Value = description.text
    sqlCommand.Parameters.Add("@linkfile", System.Data.SqlDbType.Char).Value = linkdoc.selecteditem.text
    sqlCommand.Parameters.Add("@refnumber", System.Data.SqlDbType.Char).Value = refnumber

    'delete record
    if verase.checked=true then
        querystring="DELETE calender_event where event_id=" & event_id.text
        fdate.text=""
        title.text=""
        author.text=""
        event_id.text=""
        description.text=""
        old.checked=false
        verase.checked=false
    end if 'erase.checked=true

    res=getquery(pwds)
    res.Open()
    sqlCommand.connection=res
    sqlCommand.commandtext=querystring

    try
        sqlCommand.ExecuteNonQuery()
    catch ex
        response.write("Error Updating Database")
        senderror(ex)
        res.close
        response.end
    end try

    if added=1 then 'retrieves the event id number
        querystring="SELECT max(event_id) as vid from calender_event"& _

```

```

        " WHERE refnumber=" & ref.text & " AND name=" & title.text & ""

'res.Open()
sqlcommand.commandtext=querystring

try
    datareader=sqlcommand.executereader()
catch ex
    response.write("Error obtaining key number")
    senderror(ex)
    res.close
exit sub

end try

        if datareader.read()=true then event_id.text=datareader.item("vid")
        datareader.close
        old.checked=true
        if category.selecteditem.text="Meetings" and old.checked=true then
            register.NavigateUrl="roster.aspx?event_id=" & event_id.text
            register.visible=true
        end if 'category=meetings

end if 'added=1

'clean up
res.close
sqlcommand.dispose()
sqlcommand=nothing
res.dispose()
res=nothing

End Sub 'submit button click

'view file button clicked
Sub Button2_Click(sender As Object, e As EventArgs)
    if linkdoc.selecteditem.text="None" then exit sub
    Dim root As String = server.mappath("../data/")
    Dim title as string=linkdoc.selecteditem.text
    Dim filepath As String = root & title

    If File.Exists(filepath)' And filepath.StartsWith(root) Then
        Dim filename As String = Path.GetFileName(filepath)
        Response.Clear()
        Response.ContentType = "application/octet-stream"
        Response.AddHeader("Content-Disposition", _
            "attachment; filename=""" & filename & """)
        Response.Flush()
        Response.WriteFile(filepath)
    End If 'file exists
End Sub 'button2 clicked

sub senderror(vex)

    Dim msg As MailMessage = New MailMessage()
    Smtplib.Smtplib.SmtpServer = "rialto1"
    msg.Body = vex.toString
    msg.To = "alenaburg@mercyair.net"
    msg.From = "alenaburg@mercyair.net"

```


Calendar Display

File Name: Page1.aspx

Description: Displays a table style calendar that contains hyperlinks referencing an event stored in the database.

Language: Vb.net

Code

```
<%@ Page Language="VB" Debug="true" Inherits="lifenetconn.lifenetdbconn" %>
<%@ import Namespace="System.Data" %>
<%@ import Namespace="System.Data.SqlClient" %>
<%@ import Namespace="System.IO" %>
<%@ outputcache location="none" %>
<%@ import Namespace="System.web.mail" %>
<script runat="server">

    'Event Calender Event code to populate the calender
    'By Allen Lenaburg

    Sub Page_Load(sender As Object, e As EventArgs)

        dim ex as exception
        dim vweekday as integer 'day of week that month starts on
        dim vmonthend as integer 'ending day of month
        dim ddate as date
        dim refyear as integer
        dim j as integer
        dim refdate as string
        dim prevyear as string
        dim nextyear as string

        if not ispostback then
            if request.querystring("datec")="" then
                ddate=now()
            else
                ddate="#" & format(request.querystring("datec"),"short date") & "#"
            end if 'datec=""

        else 'postback
            ddate=inidate.text
        end if 'not postback
        mth.text=monthname(month(ddate))
        yr.text=year(ddate)
        inidate.text=format(ddate,"short date")
        for j=1 to 12
            refdate=j & "/01/" & year(ddate) 'Selects month for current year
            table2.rows(0).cells(j).text="<td><a HREF=page1.aspx?datec=" & refdate & ">" & monthname(j,true) & "</a></td>"
        next

        prevyear=month(ddate) & "/01/" & year(ddate)-1
        nextyear=month(ddate) & "/01/" & year(ddate)+1
        table2.rows(0).cells(0).text="<td><a HREF=page1.aspx?datec=" & prevyear & ">{Prev Year}</a></td>"
        table2.rows(0).cells(13).text="<td><a HREF=page1.aspx?datec=" & nextyear & ">{Next Year}</a></td>"

    End Sub

End Class
```



```

'open database connection and create sqlcommand object

Dim res as sqlconnection
Dim pwds as string="n411a1" 'needed for dll file
Dim sqlCommand As New SqlCommand()

Dim datareader As sqldatareader

res=getquery(pwds)
res.open()
sqlcommand.connection=res

sqlcommand.connection=res
Dim queryString As String
querystring="select * from calender_event where month(start_dt)=" & month(ddate) & " AND year(start_dt)=" & year(ddate)
if category.selecteditem.value<>"0" then querystring=querystring & " AND category=" & category.selecteditem.text & ""
sqlcommand.commandtext=querystring

'Create and fill dataset
Dim dataAdapter As New sqlDataAdapter(sqlcommand)
Dim ds As New DataSet

try
    dataAdapter.Fill(ds,"calender_event")
catch ex
    response.write("Error Accessing Database")
    senderror(ex)
    res.close
end try
'Get weekday the month starts on and day month ends on

vweekday=GetWeekdayMonthStartsOn(ddate)
vmonthend=getdaysinmonth(month(ddate),year(ddate))

dim vcell as integer
dim vrow as integer=1
dim stri as string
dim dday as integer
dim dr as datarow
dim nn as integer
vcell=vweekday-1

for dday=1 to vmonthend
    if session("auth")<>"view" then
        stri="<font size=4><a href=event_main.aspx?evdte=" & month(ddate) & "/" & dday & "/" & year(ddate) & ">" & dday &
"</a></font>"
    else
        stri="<font size=4>" & dday & "</font>"
    end if
    'loop through dataset looking for day that matches
    for each dr in ds.tables("calender_event").rows
        if day(dr(2))=dday then
            'create a string that has day and then a hyperlink that is event
            stri=stri & "<br><a href=event_main.aspx?event_id=" & dr(0) & ">" & dr(1) & "</a>"
        end if
    next 'dr
    table1.rows(vrow).cells(vcell).text=stri 'populate cell with day and event
    vcell=vcell+1
    if vcell>6 then
        vcell=0
        vrow=vrow+1
    end if 'vcell>6
next 'dday

'clean up

```

End Sub

```

'get the weekday that the month starts on
Public Function GetWeekdayMonthStartsOn(dAnyDayInTheMonth)
    Dim dTemp as date
    dTemp = DateAdd("d", -(Day(dAnyDayInTheMonth) - 1), dAnyDayInTheMonth)
    GetWeekdayMonthStartsOn = WeekDay(dTemp)
End Function ' GetWeekdayMonthStartsOn

```

```
'Get the day the month ends on
Public Function GetDaysInMonth(iMonth, iYear)
    Dim dTemp
    dTemp = DateAdd("d", -1, DateSerial(iYear, iMonth + 1, 1))
    GetDaysInMonth = Day(dTemp)
End Function ' GetDaysInMonth
```

```
sub senderror(vex)
```

```
Dim msg As MailMessage = New MailMessage()  
SmtpMail.SmtpServer = "rialto1"  
msg.Body = vex.ToString  
msg.To = "alenaburg@mercyair.net"  
msg.From = "alenaburg@mercyair.net"  
msg.Subject = "Lifenet Error"  
SmtpMail.Send(msg)
```

End Sub 'senderror

51


```
<asp:TableCell></asp:TableCell>  
<asp:TableCell></asp:TableCell>  
<asp:TableCell></asp:TableCell>  
<asp:TableCell></asp:TableCell>  
<asp:TableCell></asp:TableCell>  
</asp:TableRow>  
<asp:TableRow Width="200px" VerticalAlign="Top" Height="100px" Font-Size="XX-Small">  
    <asp:TableCell></asp:TableCell>  
    <asp:TableCell></asp:TableCell>  
    <asp:TableCell></asp:TableCell>  
    <asp:TableCell></asp:TableCell>  
    <asp:TableCell></asp:TableCell>  
    <asp:TableCell></asp:TableCell>  
</asp:TableRow>  
<asp:TableRow Width="200px" VerticalAlign="Top" Height="100px" Font-Size="XX-Small">  
    <asp:TableCell></asp:TableCell>  
    <asp:TableCell></asp:TableCell>  
    <asp:TableCell></asp:TableCell>  
    <asp:TableCell></asp:TableCell>  
    <asp:TableCell></asp:TableCell>  
    <asp:TableCell></asp:TableCell>  
</asp:TableRow>  
</asp:Table>  
</p>  
<p>  
</p>  
<p>  
</p>  
<p>  
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
</p>  
<p>  
</p>  
<!-- Insert content here -->  
</form>  
</body>  
  
</html>
```

Search The Calendar Events

File Name: Search.aspx

Description: Displays a form that allows for user input to query a database and narrow the search for events. Events are displayed in a table with each event as a hyperlink pointing to the event maintenance page.

Language: Vb.net

Code

```
<%@ Page Language="VB" Debug="true" Inherits="lifenetconn.lifenetdbconn" %>
<%@ import Namespace="System.Data" %>
<%@ import Namespace="System.Data.SqlClient" %>
<%@ import Namespace="System.web.mail" %>
<script runat="server">

    'Calendar Search Page
    'By Allen Lenaburg

    Sub Page_Load(sender As Object, e As EventArgs)
        if not ispostback

            end if 'not post back

        End Sub

        *****
        '
        '      Data Bind
        *****
        sub databind()
            Dim ex as exception
            Dim res as sqlconnection
            Dim pwds as string="n411al" 'needed for dll file
            Dim sqlCommand As New SqlCommand()
            Dim datareader As sqldatareader
            res=getquery(pwds)
            res.Open()
            sqlCommand.connection=res
            Dim queryString As String

            queryString="SELECT * from calender_event WHERE author<>'zzzz'"
            if keyword.text<>"" then queryString=querystring & " AND name like '%" & keyword.text & "%"
            if sortby.selecteditem.value<0 then queryString=querystring & " AND category='" & sortby.selecteditem.text & ""

            'response.write(querystring)

            sqlCommand.commandtext=querystring
```


Editing The Training Roster

File Name: Roster.aspx

Description: Displays a table that lists the training roster for an event, and allows users to be added or deleted to the selected roster.

Language: Vb.net

Code

```
<%@ Page Language="VB" Debug="true" Inherits="lifenetconn.lifenetdbconn" %>
<%@ import Namespace="System.Data" %>
<%@ import Namespace="System.Data.SqlClient" %>
<%@ outputcache location="none" %>
<%@ import Namespace="System.web.mail" %>
<script runat="server">

' Roster Maint
' By Allen Lenaburg

public queryString As String 'sql query string
public refnumber as integer=0

Sub Page_Load(sender As Object, e As EventArgs)
    'populate the datagrid
    if not ispostback then
        Dim res as sqlconnection
        Dim pwds as string="n411al" 'needed for dll file
        Dim sqlCommand As New SqlCommand()
        Dim datareader As sqldatareader
        Dim querystring as string

        res=getquery(pwds)
        res.open()
        sqlCommand.connection=res

        eventid.text=request("event_id")
        linkreturn.NavigateUrl="event_main.aspx?event_id=" & eventid.text

        BindData(res,sqlcommand)
    end if 'not postback
End Sub 'page load

***** Insert new record or delete selected record *****
Sub doInsert(ByVal sender As Object, ByVal e As DataGridCommandEventArgs)

    Dim veventid as integer=cint(eventid.text)'event id
    Dim ex as exception
    'set up database connection
    Dim res as sqlconnection
    Dim pwds as string="n411al" 'needed for dll file
    Dim sqlCommand As New SqlCommand()
    Dim datareader As sqldatareader
    Dim querystring as string
```



```

        res=getquery(pwd)
res.open()
sqlcommand.connection=res

'if the Add button was clicked
If e.CommandName = "Insert" Then
    Dim txtlname As textbox
    Dim vlname As string
    Dim txtfname As textbox
    Dim vfname As string
    dim vclassid as integer
    'Read in the values of the TextBoxes
    txtlname = e.Item.FindControl("add_lname")
    vlname = txtlname.Text
    txtfname = e.Item.FindControl("add_fname")
    vfname = txtfname.Text

    querystring="INSERT INTO roster (lname,fname,classid)" & _
    " VALUES('" & vlname & "','" & vfname & "','" & veventid & "')"
    sqlcommand.commandtext=querystring

    try
        sqlcommand.ExecuteNonQuery()
    catch ex
        res.close
        response.write("Error Updating Database")
        senderror(ex)
        response.end
    end try
    'Rebind the DataGrid
    storeGrid.EditItemIndex = -1
    BindData(res,sqlcommand)

End If 'equals insert

***** Delete *****
If e.CommandName = "Delete" Then
    querystring="delete roster where id=" & e.item.cells(0).text
    'response.write(querystring)
    'response.end
    sqlcommand.commandtext=querystring
    try
        sqlcommand.ExecuteNonQuery()
    catch ex
        res.close
        response.write("Error With Updating Database")
        senderror(ex)
        response.end
    end try
    'Rebind the DataGrid
    storeGrid.EditItemIndex = -1
    BindData(res,sqlcommand)

end if 'equals delete

End Sub 'doinsert

*****Repopulates the datagrid *****
sub BindData(vres,vsqlcommand)
    Dim ex as exception
    Dim datareader As sqldatareader

    querystring="SELECT id,classid,lname,fname from roster" & _
    " WHERE classid=" & eventid.text
    vsqlcommand.commandtext=querystring

```



```

<asp:DataGrid id="storeGrid" runat="server" Width="255px" BackColor="White" BorderColor="#CCCCCC"
CellPadding="3" BorderWidth="1px" BorderStyle="None" OnItemCommand="doInsert" ShowFooter="True"
AutoGenerateColumns="False">
    <FooterStyle forecolor="#000066" backcolor="White"></FooterStyle>
    <HeaderStyle font-bold="True" forecolor="White" backcolor="#006699"></HeaderStyle>
    <PagerStyle horizontalalign="Left" forecolor="#000066" backcolor="White" mode="NumericPages"></PagerStyle>
    <SelectedItemStyle font-bold="True" forecolor="White" backcolor="#669999"></SelectedItemStyle>
    <EditItemStyle backcolor="#EEEEEE"></EditItemStyle>
    <ItemStyle forecolor="#000066"></ItemStyle>
    <Columns>
        <asp:BoundColumn Visible="false" DataField="id" ReadOnly="True" HeaderText="ID"></asp:BoundColumn>
        <asp:TemplateColumn HeaderText="Last Name">
            <ItemTemplate>
                <%# Container.DataItem("lname") %>
            </ItemTemplate>
            <FooterTemplate>
                <asp:TextBox ID="add_lname" Columns="5" Runat="Server" />
            </FooterTemplate>
            <EditItemTemplate>
                <asp:TextBox ID="lname" Columns="5" Text="<%# Container.DataItem("lname") %>" Runat="server" />
            </EditItemTemplate>
        </asp:TemplateColumn>
        <asp:TemplateColumn HeaderText="First Name">
            <ItemTemplate>
                <%# Container.DataItem("fname") %>
            </ItemTemplate>
            <FooterTemplate>
                <asp:TextBox ID="add_fname" Columns="5" Runat="Server" />
            </FooterTemplate>
            <EditItemTemplate>
                <asp:TextBox ID="fname" Columns="5" Text="<%# Container.DataItem("fname") %>" Runat="server" />
            </EditItemTemplate>
        </asp:TemplateColumn>
        <asp:TemplateColumn>
            <ItemTemplate>
                <asp:Button CommandName="Delete" BackColor="#E0E0E0" Text="X" ForeColor="Red" ID="btnDel"
Runat="server" />
            </ItemTemplate>
            <FooterTemplate>
                <asp:Button CommandName="Insert" BackColor="#E0E0E0" Text="Add" ForeColor="DarkBlue" ID="btnAdd"
Runat="server" />
            </FooterTemplate>
        </asp:TemplateColumn>
    </Columns>
</asp:DataGrid>
</div>
<div align="center">
</div>
<div align="center">
</div>
<div align="center">
</div>
</form>
</body>
</html>

```

Calendar For Formatted Date Entry

File Name: Calendar.html

Description: Once a user clicks the select date button a small calendar appears allowing for date selection. The selected date is then formatted and placed in the correct text box.

Language: Javascript

Code

```
<%
response.expires = 0
%>
<html>
<head>
<title>Calendar</title>
<!-- Used to create Popup calendar to choose a date -->
<!-- This code can be reused as long as this copyright notice is not removed -->
<!-- Copyright 1999 InsideDHTML.com, LLC. All rights reserved.
See www.siteexperts.com for more information.
-->
<style type="text/css">
    .today {color:navy; font-weight:bold}
    .days {font-weight:bold}
</style>
<script language="Javascript">
    // Initialize arrays.
    var adate=dialogArguments;

    var months = new Array("January", "February", "March",
        "April", "May", "June", "July", "August", "September",
        "October", "November", "December");
    var daysInMonth = new Array(31, 28, 31, 30, 31, 30, 31, 31,
        30, 31, 30, 31);
    var days = new Array("Sunday", "Monday", "Tuesday",
        "Wednesday", "Thursday", "Friday", "Saturday");

    function getDays(month, year) {
        // Test for leap year when February is selected.
        if (1 == month)
            return ((0 == year % 4) && (0 != (year % 100))) ||
                (0 == year % 400) ? 29 : 28;
        else
            return daysInMonth[month];
    }

    function getToday() {
        // Generate today's date.
        if (adate != "")
        {
            this.now = new Date(adate);
        }
        else
```

```

    {
    this.now = new Date()
    }
    this.year = this.now.getFullYear();
    this.month = this.now.getMonth();
    this.day = this.now.getDate();
}

// Start with a calendar for today.
today = new getToday();

function newCalendar() {
    today = new getToday();
    var parseYear = parseInt(document.all.year
    [document.all.year.selectedIndex].text);
    var newCal = new Date(parseYear,
    document.all.month.selectedIndex, 1);
    var day = -1;
    var startDay = newCal.getDay();
    var daily = 0;
    if ((today.year == newCal.getFullYear()) &&
    (today.month == newCal.getMonth()))
        day = today.day;
    // Cache the calendar table's tBody section, dayList.
    var tableCal = document.all.calendar.tBodies.dayList;
    var intDaysInMonth =
    getDays(newCal.getMonth(), newCal.getFullYear());
    for (var intWeek = 0; intWeek < tableCal.rows.length;
    intWeek++)
        for (var intDay = 0;
            intDay < tableCal.rows[intWeek].cells.length;
            intDay++) {
            var cell = tableCal.rows[intWeek].cells[intDay];

            // Start counting days.
            if ((intDay == startDay) && (0 == daily))
                daily = 1;

            // Highlight the current day.
            cell.className = (day == daily) ? "today" : "";

            // Output the day number into the cell.
            if ((daily > 0) && (daily <= intDaysInMonth))
                cell.innerText = daily++;
            else
                cell.innerText = "";
        }
    }

function getDate() {
    var sDate;
    // This code executes when the user clicks on a day
    // in the calendar.
    if ("TD" == event.srcElement.tagName)
        // Test whether day is valid.
        if (" " != event.srcElement.innerText)
            //alert(event.srcElement.innerText);
            sDate = document.all.month.value + "/" + event.srcElement.innerText + "/" +
document.all.year.value;
            document.all.ret.value = sDate;
            window.close();
    }
}
</script>

```

```

</head>
<body onload="newCalendar()" onunload="window.returnValue = document.all.ret.value;">
  <input type="hidden" name="ret" />
  <table id="calendar">
    <thead>
      <tr>
        <td align="middle" colspan="7">
          <!-- Month combo box -->
          <select id="month" onchange="newCalendar()">
            <script language="Javascript">
              // Output months into the document.
              // Select current month.
              for (var intLoop = 0; intLoop < months.length;
                intLoop++)
                document.write("<OPTION VALUE=" + (intLoop + 1) + " " +
                  (today.month == intLoop ?
                    "Selected" : "") + ">" +
                    months[intLoop]);
            </script>
          </select>
          <!-- Year combo box -->
          <select id="year" onchange="newCalendar()">
            <script language="Javascript">
              // Output years into the document.
              // Select current year.
              for (var intLoop = today.year; intLoop < (today.year + 4);
                intLoop++)
                document.write("<OPTION VALUE=" + intLoop + " " +
                  (today.year == intLoop ?
                    "Selected" : "") + ">" +
                    intLoop);
            </script>
          </select>
        </td>
      </tr>
      <tr class="days">
        <!-- Generate column for each day. -->
      </tr><script language="Javascript">
        // Output days.
        for (var intLoop = 0; intLoop < days.length;
          intLoop++)
          document.write("<TD>" + days[intLoop] + "</TD>");
      </script>
    </thead>
    <tbody id="dayList" onclick="getDate()" align="middle">
      <!-- Generate grid for individual days. -->
      <script language="Javascript">
        for (var intWeeks = 0; intWeeks < 6; intWeeks++) {
          document.write("<TR>");
          for (var intDays = 0; intDays < days.length;
            intDays++)
            document.write("<TD></TD>");
          document.write("</TR>");
        }
      </script>
    </tbody>
  </table>
  <center>
    <input style="WIDTH: 50pt" onclick="Cancel();" type="button" value="Cancel" />
  </center>
  <script language="Javascript1.2">

    function Cancel() {
      document.all.ret.value = "";
      window.close();
    }
  </script>

```

```
</script>  
</body>  
</html>
```

Document Publishing Page

File Name: Docpublish.aspx

Description: Displays an upload button that allows users to upload to the server with limitations of file types. The file database is then updated with the file information

Language: Vb.net

Code

```
<%@ Page Language="VB" Debug="true" Inherits="lifenetconn.lifenetdbconn" %>
<%@ import Namespace="System.Data" %>
<%@ import Namespace="System.Data.SqlClient" %>
<%@ import Namespace="System.IO" %>
<%@ import Namespace="System.web.mail" %>
<script runat="server">

'Document Publish
'By Allen Lenaburg
    public refnumber as integer
    Sub Page_Load(sender As Object, e As EventArgs)

End Sub

'*****
'Upload Procedure
'*****
sub fileupload(sender As Object, e As EventArgs)
dim exa as exception

Dim directorySeparatorChar As Char = Path.DirectorySeparatorChar
dim currentDir as string=server.mappath("../data/")
If Not (uploadedFile.PostedFile Is Nothing) Then
    dim cleared as string="no"

    Dim postedFile = uploadedFile.PostedFile
    Dim filename As String = Path.GetFileName(postedFile.filename)
    if len(filename)>49 then
        refnumber=70
        exit sub
    end if
    select case right(filename,3)
        case "doc","xlw","xls","ppt","txt","zip","pdf","htm"
            cleared="yes"
    end select

    select case right(filename,4)
        case "html"
            cleared="yes"
    end select

    if cleared="no" then
        refnumber=50
        exit sub
    end if
end if
```



```

Dim contentType As String = postedFile.ContentType

Dim contentLength As Integer = postedFile.ContentLength
try
    postedFile.SaveAs(currentDir & _
        directorySeparatorChar.ToString() & filename)
Catch exa
    cleared="no"
    refnumber=80
End Try
*****
'Append or Update database
*****

'database_connection()
if cleared="no" then exit sub
Dim res as sqlconnection
Dim pwds as string="n411al" 'needed for dll file
Dim sqlCommand As New SqlCommand()
Dim datareader As sqldatareader
Dim querystring as string
Dim ex as exception
res=getquery(pwds)
res.open()
sqlcommand.connection=res

querystring="select [filename] from fload where [filename]='" & filename & "'"

sqlcommand.commandtext=querystring
datareader=sqlcommand.executereader
if datareader.read()=true then
    datareader.close()

    querystring="UPDATE fload set filename=@filename"& _
        ",category=@category"& _
        ",author=@author"& _
        ",description=@description"& _
        ",lockout=@lockout,Datecreated=@timestamp"& _
        " where [filename]='" & filename & "'"
else
    datareader.close()
    querystring="INSERT INTO fload (filename,category,description,author,lockout,Datecreated)"& _
        " Values(@filename,@category,@description,@author,@lockout,@timestamp)"
end if 'datareaser=true

sqlCommand.Parameters.Add("@filename", System.Data.SqlDbType.Char).Value = filename
sqlCommand.Parameters.Add("@category", System.Data.SqlDbType.Char).Value = category.selecteditem.value
sqlCommand.Parameters.Add("@description", System.Data.SqlDbType.Char).Value = description.text
sqlCommand.Parameters.Add("@author", System.Data.SqlDbType.Char).Value = author.text
sqlCommand.Parameters.Add("@lockout", System.Data.SqlDbType.bit).Value = lockout.checked
sqlCommand.Parameters.Add("@timestamp", System.Data.SqlDbType.Char).Value = now()
sqlcommand.commandtext=querystring

try 'execute the command
    sqlcommand.executenonquery
catch ex
    refnumber=40
    senderror(ex)
finally
    res.close

```

```

sqlcommand.dispose()
sqlcommand=nothing
res.dispose()
res=nothing
datareader=nothing
end try

End If 'uploaded file not nothing

end sub 'uploadclick

*****

sub senderror(vex)

Dim msg As MailMessage = New MailMessage()
    SmtMail.SmtpServer = "rialto1"
    msg.Body = vex.toString
    msg.To = "alenaburg@mercyair.net"
    msg.From = "alenaburg@mercyair.net"
    msg.Subject = "Lifenet Error"
    SmtMail.Send(msg)
End Sub

</script>
<html>
<head>
    <script language="Javascript">

        function f(i)
        {
            if(i==40)
            {
                alert('A Database Error Occurred Upload Not Completed');
            }
            if(i==50)
            {
                alert('Error: Invalid File Type');
            }

            if(i==70)
            {
                alert('Error: Filename Too Long');
            }

            if(i==80)
            {
                alert('Error: Fileupload error');
            }

        } //function f(i)

    </script>
</head>
<body onload="Javascript:f(<%=refnumber%>);">
    <form runat="server">
        <p align="center">
            <font size="5">Document Publishing</font>
        </p>

```

[illegible]

Document Viewing Page

File Name: Docview.aspx

Description: Displays a list of files that are available for download from the server. Also allows for file deletion and file locking.

Language: Vb.net

Code

```
<%@ Page Language="VB" Debug="true" Inherits="lifenetconn.lifenetdbconn" %>
<%@ import Namespace="System.Data" %>
<%@ import Namespace="System.Data.SqlClient" %>
<%@ import Namespace="System.IO" %>
<%@ outputcache location="none" %>
<%@ import Namespace="System.web.mail" %>
<script runat="server">
```

*'Document View
'By Allen Lenaburg*

```
Sub Page_Load(sender As Object, e As EventArgs)
```

```
    if session("auth")="view" then dg1.columns(4).visible=false
        Dim x as integer
        if not ispostback
            bindata()
        end if 'not a postback
    End Sub
```

```
Sub dg1edit(source As Object, e As DataGridCommandEventArgs)
    'responds to either delete or download button clicks
    if e.commandname="Delete" then delete_file(e.item.cells(1).text)
    if e.commandname="Download" then download_files(e.item.cells(1).text)
end sub
```

```
'Delete File
Sub delete_file(filepath)
    Dim res as sqlconnection
        Dim pwds as string="n411al" 'needed for dll file
        Dim sqlCommand As New SqlCommand()
        Dim datareader As sqldatareader
```

```
    Dim ex as exception
    Dim root As String = server.mappath("../data/")
```

```
    If File.Exists(root & filepath) And filepath.StartsWith(root) Then
        try
            File.Delete(root & filepath )
```

```

catch ex
    response.write("Error Deleting File")
    senderror(ex)
    response.end
end try
end if

res=getquery(pwds)
res.Open()
sqlcommand.connection=res
Dim qrystring as string="DELETE fload where [filename]='" & filepath & "'"
sqlcommand.commandtext=qrystring

try
    sqlcommand.ExecuteNonQuery
catch ex
    res.close
    response.write("Error Deleting From Database")
    senderror(ex)
    response.end
end try
    sqlcommand.dispose()
    sqlcommand=nothing
    res.dispose()
    res=nothing
'end try
bindata()
end sub

*****
'Download files
*****

sub download_files(title)

    Dim root As String = server.mappath("../data/")
    Dim filepath As String = root & title

    If File.Exists(filepath) And filepath.StartsWith(root) Then
        Dim filename As String = Path.GetFileName(filepath)
        Response.Clear()
        Response.ContentType = "application/octet-stream"
        Response.AddHeader("Content-Disposition", _
            "attachment; filename=""" & filename & """)
        Response.Flush()
        Response.WriteFile(filepath)
    End If

end sub

***** Bind Data *****
sub bindata()
    'label6.text=category.selecteditem.value
    dim ex as exception
    Dim res as sqlconnection
        Dim pwds as string="n411al" 'needed for dll file
        Dim sqlCommand As New SqlCommand()
        Dim datareader As sqldatareader
    res=getquery(pwds)
    res.Open()
    sqlcommand.connection=res
    Dim queryString As String
    qrystring="select * from fload"
    select case sortby.selecteditem.value

```


Manage User Accounts

File Name: Gatemaint.asp

Description: Lists user account information and allows modifying user's passwords and privileges.

Language: Vbscript

Code

```
<%
'Account Maint

'By Allen Lenaburg

if session("master")<>true then response.redirect "2def.htm"
' declare variables.
dim cnn
dim rst
dim vlname
dim vfname
dim vusername
dim vpwd
dim vcategory
dim vid
Dim lb_validData
'on error resume next
' start writing the html document
With Response
    .Write "<html>"
    .Write "<head>"
    .Write "<title>Gate Maint</title>"
    .Write "<LINK rel='stylesheet' type='text/css' href='default.css'>"
    .Write "</head>"
    .Write "<body>"
End With
' open the database object
' do the work
If ( Request.Form("ls_submit") = "Submit" ) then ' save the form data
' delete routine
if request.form("ls_delete")="on" then
    llreturn=deleterecord()
    response.end
end if

    ll_return = save() ' call the local function to save the record
    if ll_return>0 then response.write "Error " & ll_return
    response.end
response.write err.description
else
' try to pull up the record from the UID
    vuserid=request("q")
    If vuserid <> "" Then ' pull up the event_id's record
        ' get the record from the database
        ' open the database object
        opendatabase cnn,rst
        sql="select * from passgate where id=" & vuserid
```



```

rst.open(sql),cnm,2,3
' put the recordset's values into local variables
    If not rst.EOF Then
        ' read in the strings
        vlname=rst("lname")
        vfname=rst("fname")
        vusername = rst( "username" )
        vpwd = rst( "pwd" )
        vcategory = rst( "category" )
        vid=rst("id")
    end if
    'close database
    rst.close
    cnm.close
    set rst=nothing
    set cnm=nothing

end if

End If

' Write out the form
With Response
    ' start writing out the form
    .Write "<form name=form1 method=post>"
    .Write "<br>"
    .Write "<h2>View/Edit Data</h2><br>"
    .write "Record " & vid
    .Write "<table border=0 width=50%>"
    .Write "<tr align='left'>"
    .Write "<td class='eventLabel' valign='top'>Last Name:</td>"
    .Write "<td><input type='text' name='lname' value='" & vlname & "'></td>"
    .Write "<td class='eventLabel' valign='top'>First Name:</td>"
    .Write "<td><input type='text' name='fname' value='" & vfname & "'></td></tr>"
    .Write "<tr><td class='eventLabel' valign='top'>Username:</td>"
    .Write "<td><input type='text' name='username' value='" & vusername & "'></td>"
    .Write "<td class='eventLabel' valign='top'>Password:</td>"
    .Write "<td><input type='text' name='pwd' value='" & vpwd & "'></td></tr>"
    .Write "<td class='eventLabel' valign='top'>Category:</td>"
    .Write "<td><input type='text' name='category' value='" & vcategory & "'></td>"
    .write "<td><input type='hidden' name='uid' value='" & vid & "'></td>"
    .Write "</tr>"
    .Write "<tr align='left'>"

    ' Write out the submit button and close
    .Write "<td class='eventLabel' valign='top'>Delete:</td>"
    .Write "<td colspan=3 align='left'><input type='checkbox' name='ls_delete' ></td></tr>"
    .Write "<tr><td colspan=3 align='right'><input type='submit' name='ls_submit' value='Submit'></td>"
    .Write "</tr>"
    .Write "</table>"
    .Write "<br><br>"
    .Write "</form>"

    .write "<input type='button' name='History' value='History'"
    .write "onclick=document.location.href='loghistory.ASP?q=" & vid & "'>"
    .write "<input type='button' name='Return' value='Return'"
    .write "onclick=document.location.href='gatelists.ASP'>"
    .Write "</body>"
    .Write "</html>"

End With
response.write err.description
*****
Private Function save()
    ' declare some variables
    Dim cnm
    Dim rst

```

```

Dim sql

' get the existing record, or get a new record.
vusername= Request("username")
sql = "SELECT * FROM passgate WHERE username='" & vusername & "'",

'Open database
opendatabase cnn,rst
with rst
    .Open sql, cnn,2,3
    If .EOF Then .AddNew ' if there wasn't a record, add a new one

    ' set values for the fields in the record
    .Fields("lname")=request.form("lname")
    .Fields("fname")=request.form("fname")
    .Fields("username") = Request.Form("username")
    .Fields("pwd") = Request.Form("pwd")
    .Fields("category") = Request.Form("category")
    .Update
    .close
End With

' close and dispose of resources
cnn.close
Set rst = Nothing
response.write "<br>Record " & request.form("username") & " Saved</br>"
response.Write "<br><a href='gatelist.asp'>Return</a></br>"

End Function ' save
*****

private function deleterecord()
dim vusername
vuserid=request.form("uid")

sql = "delete FROM passgate WHERE id=" & vuserid & ","
'open database
opendatabase cnn,rst
cnn.execute(sql)
cnn.close
set cnn=nothing
response.write "<br>User " & request.form("uid") & " Is Deleted</br>"
response.Write "<br><a href='gatelist.asp'>Return</a></br>"
response.write err.description
response.end
end function
*****

sub opendatabase(fcnn,first)
set fcnn=server.createobject("adodb.connection")
set first=server.createobject("adodb.recordset")
fcnn.open "dsn=mercy.sitedata;uid=admin;pwd=n411al"
'first.open "passgate",cnn,2,3
'sql="create table passgate (username varchar(20) not null,"
'sql=sql & "pwd varchar(10) not null,category varchar(10) not null)"
'cnn.execute (sql)
end sub
*****

%>

```

Display User Activity

File Name: Loghistory.asp

Description: Displays a history of user's access to the intranet site.

Language: Vbscript

Code

```
<html>
<head>

</head>

<body>

<p align="center">LOG HISTORY</p>

<%
dim vuser
'Displays the history of the selected person
'By Allen Lenaburg
if session("master")<>true then response.redirect "2def.htm"
response.write "<p><center>User Log</center></p>"
response.write "<br>"
dim cnn
dim rs
vuser=request("q")
set cnn=server.createobject("adodb.connection")
set rs=server.createobject("adodb.recordset")
cnn.open"dsn=mercy.sitedata;uid=admin;pwd=n411al"
'sql="select * from passlog order by id"
sql="SELECT passgate.id,passgate.username,passgate.lname, passgate.fname, passlog.sessiondate"
sql=sql & " FROM passgate INNER JOIN passlog ON passgate.id = passlog.staffid"
sql=sql & " where [passgate].[id]=" & vuser
sql=sql & " ORDER BY datevalue(passlog.sessiondate);"
rs.open(sql),cnn,2,3
if rs.bof=true or rs.eof=true then
    response.write "No Records"
    call database_close(cnn,rs)
else
rs.movefirst
with response
.write "<p><center>[" & rs("id") & "]"&nbsp;&nbsp;&nbsp;& rs("fname") & "&nbsp;&nbsp;&nbsp;& rs("lname") & "<p>"
.write "<table>"
do while not rs.eof
    .write "<tr><td>&nbsp;&nbsp;& rs("sessiondate") & "</td></tr>"
rs.movenext
loop
.write "</table>"

end with
end if
end %>
```

```
*****  
function database_close(fcnn,frs)  
frs.close  
set frs=nothing  
fcnn.close  
set fcnn=nothing  
end function  
  
%>  
</body>  
  
</html>
```

User Authentication Page

File Name: passgate.asp

Description: Authenticates users against a database and assigns the appropriate privilege levels.

Language: Vbscript

Code

```
<%
on error resume next
dim cnn
dim rs
dim vid
dim vlname
dim vfname
dim vuser
dim vpassw
vuser=lcase(request.form("user"))
vpassw=lcase(request.form("passw"))
set cnn=server.createobject("adodb.connection")
set rs=server.createobject("adodb.recordset")
cnn.open "dsn=mercy.sitedata;uid=admin;pwd=n411al"
sql="select * from passgate "
sql=sql + "where username='" & vuser & "' and pwd='" & vpassw & "'"
rs.open (sql),cnn,2,3
if not rs.eof then
select case rs("category")
case "Master"
session("master")=true
session("authenticated")=true
session("userauth")=true
case else
response.redirect "2def.htm"
end select
else
response.redirect "2def.htm"
end if
vid=rs("id")' ****set session id for list boxes *****
session("sid")=vid
vid=rs("id")
vdate=cstr(now)
sql="insert into passlog (staffid,sessiondate)"
sql=sql & "values ('" & vid & "','" & vdate & "')"
cnn.execute(sql)
set frs=nothing
rs.close
set rs=nothing
cnn.close
set cnn=nothing
response.redirect "maingate.htm"
response.write err.description & "<p>"
%>
```

User Log-in Page

File Name: 2def.htm

Description: Form that permits users to enter username and password information to be authenticated.

Language: HTML

Code

[illegible]

Database Connection File

File Name: Vbconnect.vb

Description: File that contains the information for connecting to a database. Should be compiled by the utility vbc.exe.

Language: vb.net

Code

*'Database Connection
'By Allen Lenaburg*

```
Imports System
imports system.data
imports system.data.SqlClient
imports system.xml
```

```
Namespace conn
```

```
    Public Class dbconn
    inherits System.Web.UI.Page
    Function getquery(pwds)
    dim ex as exception
    try
    if pwds="password!" then
```

```
Dim connectionString as String="server=myserver'; user id='user id'; password='mypassword'; Database='mercya'"
```

```
    Dim sqlConn As New SqlConnection(connectionString)
    Return sqlconn
    else
    response.write("Not Permitted")
    end if
```

```
    catch ex
    response.write("Error: " & ex.message)
    end try
    End Function
End Class
End Namespace
```


Sample Connection To A Microsoft Access Database

File Name: Odbc.aspx

Description: Contains the information for connecting to a Microsoft Access database.

Language: Vb.net

Code

'Sample Code For Access Database

'By Allen Lenaburg

```
Sub Page_Load(sender As Object, e As EventArgs)
Dim oODBCConnection As Odbc.OdbcConnection
Dim sConnString As String = "Dsn=mercya.sitedata;" & _
    "Uid=userid;" & _
    "Pwd=password"
oODBCConnection = New Odbc.OdbcConnection(sConnString)
oODBCConnection.Open()

Dim cmd As New OdbcCommand()
cmd.connection=oODBCConnection
Dim queryString As String
Dim datareader As OdbcDataReader
queryString="select * from roster"
cmd.commandtext=querystring
datareader=cmd.ExecuteReader()

datareader.read()
response.write(datareader.item(1))

oODBCConnection.close
cmd.dispose()
cmd=nothing
oODBCConnection.dispose()
oODBCConnection=nothing
End Sub
```

APPENDIX D

DATABASE STRUCTURE

Table 1

Calendar Events

Field Name	Type	Description
Event_id	Integer	Primary Key
Name	Text	Title of event
Start_dt	Date	Starting date
Start_time	Date	Starting time
End_dt	Date	Ending date
End_time	Date	Ending time
Location	Text	Event location
Description	Text	Event description
Lastmodified_dt	Date	Timestamp
Category	Text	Event category
Author	Text	Event creator

Note. Table name is event_calendar and stores data related to events.

Table 2

User Accounts

Field Name	Type	Description
Id	Integer	Primary Key
Username	Text	Employee username
Pwd	Text	Employee password
Category	Text	Privilege level
Lname	Text	Last name
Fname	Text	First name

Note. Table name is passgate and contains user account information

Table 3

User Log-in History

Field Name	Type	Description
Id	Integer	Primary Key
Username	Text	Employee username
Sessiondate	Date	Date of session
Staffid	Integer	Foreign key linked to passlog primary key

Note. Table name is passlog and records user activity

Table 4

Training Roster

Field Name	Type	Description
Id	Integer	Primary key
Classid	Integer	Foreign key linked to Event_calendar primary key
Employeeid	Integer	Foreign key linked to passlog primary key
Email	Text	User Email
Phone	Text	User Telephone
Confirmed	Yes/no	Confirmation of attendance
Today_date	Date	Date of registration

Note. Table name is roster, and it maintains training rosters for selected events

Table 5

File Publishing

Field Name	Type	Description
Id	Integer	Primary key
Filename	Text	Name of file
Category	Text	Category of file
Description	Text	Description of file
Author	Text	File creator
Lockout	Yes/no	File lock status
Datecreated	Date	Date uploaded

Note. Table name is fload, and it tracks files uploaded and viewed on the server

REFERENCES

- Davis, F. (1989). Percieved Usefulness, Percieved Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*. September, 318-340.
- Deimler, M., Hansen, M. (2001). Cutting costs while improving morale with B2E management. *MIT Sloan Management Review*. 43, 96-100.
- Devendra, G., (1997). Intranets: the second wave. *Management Accounting*, January, 6-9.
- Durval, M., (2002). Hosted intranets proving their worth. *Intranet.com*, Retrieved April 25, 2003, from http://intranetjournal.com/articles/pic_30_10_01a.html.
- Fulk, J., Hinds, P., Hollingshead, F., Kiesler, S., & Monge, P. (2002). Fostering intranet knowledge sharing: An integration of transactiave memory and public goods approaches. *A Distributed Work*. Massachusetts Institute of Technology.
- Guengerich, S., Graham, D., Miller, M., & McDonald, S. (1997). *Building The Corporate Intranet*. NY: Wiley.
- Jones, M. (2000). Small business gets the web. *Internet Week*, 807, 66-67.
- McGaughey, R. (1999). Internet technology: contributing to agility in the twenty-first century. *International Journal of Agile Management Systems*. 1, 7-13.
- Pitt, L., Murgolog-Poore, M., Dix, S. (2001). Changing change management: The intranet as catalyst. *Journal of Change Management*. London, 2, 106-114.
- Roberts, John. (2002). Forecast: Small businesses up IT spending priorities. *The Newsweekly for Builders of Technology Solutions*. 1020, A24.
- Robertson, J. (2002), Keeping your intranet healthy and effective. *Step Two Designs*. Retrieved April 25, 2003, from <http://www.steptwo.com.au/papers/>

Sanders, T. (1999). Turning intranets into a profitable business. *Computer Reseller News*, 840, 81-87

Smith, I. Sharing information in your company: is it time for an intranet?. (2001). *Intranet Journal*. Retrieved April 25, 2003, from http://intranetjournal.com/articles/pic_10_31_01a.html.

Starnes, J. (1997). *The complete intranet source for information professionals*. Kent, OH: Special Library Association.

Tittel, E. (1997). *Intranet Bible*. Foster City, CA.: IDG Books.

Ward, T. (2001). Five winning intranet characteristics. *Intranet Journal*. Retrieved April 25, 2003, from http://intranetjournal.com/articles/pin_09_19_01a.html.

White, M. (2001). Avoiding black hole intranets. *Econtent*, 4,24, 50-51.